

**PRUEBA N°1 ESTRUCTURA DE DATOS
PRIMER SEMESTRE 2006**

NOMBRE:..... SECCIÓN:.....

PARTE I: Marcar la opción que corresponda (3*buenas - malas)

1. Que problema tiene el siguiente código:

```
int *p = new int[100];  
delete p;
```

- a) Mal definición de p.
- b) Solo borra el elemento p[0] y el resto queda sin "acceso";
- c) Borra todos los elementos del arreglo.
- d) Ninguna de las Anteriores.

2. Dada la siguiente declarativa

```
class Algo{  
    public:  
        float tonto(int a) { return a * 2.0; }  
};  
Algo o;
```

Cual es la opción incorrecta?

- a) int x = tonto(1);
- b) float r = o.tonto(3);
- c) cout << o.tonto(2);
- d) Ninguna de las Anteriores.

3. Teniendo el siguiente código:

```
int entero;  
float real;  
  
void hola(int a, float r, float& a, int *p);
```

Cual es la llamada válida?

- a) hola(real, entero, &real, &entero);
- b) hola(entero, real, real, &entero);
- c) hola(entero, real, &real, entero);
- d) Ninguna de las Anteriores.

4. Qué valor se imprime?

```
int i = 3, j = 6;  
float r = 7.0;  
  
int cosa(int t, int *a) { return t * (*a);}  
int cosa(float t, int *a) { return t + *a; }  
int cosa(double t, int *a) { return t / *a;}  
  
cout << cosa(r, &i);
```

- a) 21
- b) 10
- c) 2.33
- d) Ninguna de las Anteriores.

5. Dada la siguiente declarativa:

```
class P
{
    public:
        int suma(int a) { return a + m; }
        int n;
    private:
        int m;
};

P o;
```

Que sentencia es incorrecta?

- a) o.m = 3;
- b) o.n = 9;
- c) o.suma(3);
- d) Ninguna de las Anteriores.

6. Se dispone de la siguiente declaración

```
template <class T, class U>
T funcion(U p)
{
    return (T) p;
}
```

Son correctas?

- I) int x = funcion(3.14127312);
- II) float r = funcion(3);
- III) cout << funcion(2);

- a) Solo I y III
- b) Solo II
- c) Solo III
- d) Ninguna de las Anteriores.

7. Dada la siguiente clase:

```
class Padre
{
    public:
        void imprimir();
}
```

```
        private:  
            int variable;  
};
```

Para crear la clase Hijo, que herede todo de la clase Padre, que sentencia se debe usar?

- a) class Hijo : public Padre {};
- b) class Padre : public Hijo {}
- c) class Hijo(Padre) {}
- d) Ninguna de las Anteriores.

8. Dada la declaración:

```
class A
{
    public:
        int f(int a)
        {
            B b;
            return a + b.f(a);
        }
};

class B
{
    public:
        int f(int b)
        {
            C c;
            return (b + 1) * c.f(b / 2);
        }
};

class C
{
    public:
        int f(int r)
        {
            return r * r;
        }
};

A a;
cout << a.f(3);
```

Que valor se imprime?

- a) No imprime nada
- b) 3
- c) 7
- d) Ninguna de las Anteriores.

9. Dada la siguiente declarativa

```
Class Numero
{
    public:
        int doble();
}
```

```
        private:  
            int base;  
    }
```

Que implementación es la correcta para definir la función miembro "doble".

- a) int doble() { return base + base}
- b) int Numero::doble() { return base + base}
- c) Numero doble() { return base + base}
- d) Ninguna de las Anteriores.

10. Dadas las siguientes declaraciones:

```
I) int sub(int j, float r, int g = 0)
II) float sub(int j = 0, float t, int r = 0)
III) void sub(int j = 0, float t = 5, int w = 0)
```

Cúal o cuales son declaraciones correctas?

- a) Solo I
- b) I y III
- c) Todas
- d) Ninguna de las Anteriores.

PARTE II: (15 puntos)

Se tiene una Cola con dígitos (0..9). Se pide implementar el operador recursivo **Numero(Q)** que permite retornar el número formado por los dígitos de la cola Q

Por ejemplo si se realizan las siguientes operaciones sobre la cola Q

```
Q.agregar(1);
Q.agregar(2);
Q.agregar(3);
Q.agregar(4);
```

El operador Numero(Q) debe retornar 4321

PARTE III (15 puntos)

Se dispone de un stack de números enteros. Se pide escribir el operador **Separa(S)**, que permite separar los elementos de S de tal forma que los números pares queden en la parte inferior del stack ordenados en forma descendente; y en la parte superior del stack deben quedar los elementos impares ordenados en forma ascendente. Puede utilizar como estructuras auxiliar algún TAD.

AUTA PARTE II

1 b
2 a
3 b
4 b
5 a
6 d
7 c
8 a
9 b
10 b

PAUTA PARTE II

```
template <class T>
T numero(ColaAbstracta<T> &q)
{
    if(q.vacia()) {
        return 0;
    }
    else {
        return q.extraer() + numero(q) * 10;
    }
}
```

PAUTA PARTE III

```
template <class T>
void separa(PilaAbstracta<T> &p)
{
    ColaDinamica<T> q;
    PilaDinamica<T> t;
    T x;
    while(!p.vacio()) {
        x = p.pop();
        if(x % 2 == 0)
            q.agregar(x);
        else
            t.push(x);
    }
    while(!q.vacia()) {
        p.push(q.extraer());
    }
    while(!t.vacio()) {
        p.push(t.pop());
    }
}
```